International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 05 Issue: 05 | May-2018 www.irjet.net p-ISSN: 2395-0072

Performance Evaluation of Cloud and Mobile Application

Vijaya Shetty S[1], Sarojadevi H[2]

[1] *Assoc. Professor, Department of Computer Science & Engineering, Nitte Meenakshi Institute of Technology Bengaluru, India-560064*
[2] *Professor, Department of Computer Science & Engineering, Nitte Meenakshi Institute of Technology Bengaluru, India-560064*

Abstract - *Today we live in the era of cloud and mobile computing which has brought a scientific and technological revolution in the mobile app market. This has also brought a swift change in all spheres of life, since mobile apps using cloud services are becoming an integral part of this technological system. To maintain and improve the quality of their apps, developers need data about how their app is performing in the wild. Performance analysis of applications helps identify where and how our application can benefit from available hardware resources. The asynchronous, multi-threaded nature of mobile apps makes tracing difficult. The difficulties are compounded by the resource limitations inherent in the mobile platform. To address this challenge a multi-threaded transactional Android application similar to Amazon book store is developed and its transactional performance is analysed. As multi-threaded transactional applications are widely used in enterprise applications, the needs for systematic performance analysis of such applications have significantly increased. Dalvik Debug Monitoring System (DDMS), New Relic and Jmeter profilers/tools have been used for measuring performance metrics of significant interest at different layers of the application.*

***Key Words*: Cloud, Mobile, Performance, DDMS, New-Relic, Android**

1. INTRODUCTION

Cloud Computing is a fast growing technology in the computing world. Mobile Cloud Computing integrates the cloud computing concept with a mobile environment. Mobile Cloud Computing eliminates some of the performance issues associated with mobile devices. For sustained and improved quality of mobile applications, developers need to know about the performance of the application developed. Performance analysis helps to leverage the benefits of available hardware resources.

Smart devices such as smartphones and tablets use iOS and Android OS. Android is open source free platform which the developers can use to develop their own applications. The actual performance of application varies with hardware/software components used in the design. Developers struggle to achieve higher optimized performance at lower cost.

Users rely on various apps in app marketplace for a variety of tasks such as posting comments on social networks to banking. To enhance the performance of apps, developers need to know artifact performance of apps.

The mobile-application marketplace is highly competitive and growing day by day. Tracing of mobile apps is difficult because these apps are asynchronous and multithreaded in nature. The difficulties become more intense due to the inherent resource limitations of the mobile platform [1]. The basic problems that occur in mobile apps are responsiveness, storage, connectivity, usability, CPU performance and reliability.

As a study of this challenge, a multi threaded Android transactional application; Scalibs(Scalable Libraries) has been implemented and analyzed for the performance of transactions in different perspective. DDMS, New Relic and Apache JMeter tools have been used for measuring performance metrics of the application.

2. RELATED WORK

The need for systematic performance analysis of Android platform is increased significantly as this platform is widely used in smart mobile devices. Performance of computer systems is usually measured using benchmark applications and profiler software [2]. Hyeon-Ju Yoon [3] studied on the performance of Android platform using a benchmark application and public profile software. For more detailed and integrated performance analysis, authors proposed a profiling architecture of Android platform.

The Android platform does not have proper performance tool for evaluating and monitoring apps in a consistent and systematic way. As a key contribution, Xuetao Wei et Al.[4], designed and implemented ProfileDroid, a multi-layer profiling and monitoring system for mobile android apps. Their approach profiles apps at different levels; which includes static, user interaction, OS, and network levels. Approximately 27 Android apps have been analyzed by them. Their observation based analysis includes the analysis of app execution, network traffic, security of traffic and apps communication.

The Android based applications in smartphones generate IO requests of unique type. Current workload generators and profiling tools are not designed to generate and capture the Android apps' IO requests. The Android storage performance analysis Tool (AndroStep) is designed for analyzing the IO subsystem behavior in devices based

on Android [5]. AndroStep consists of workload generator, called Mobibench, and workload analyzer, called Mobile Storage Analyzer Tool (MOST). Mobibench is an android App, that generates typical file system workloads, e.g., Random vs. Sequential and Synchronous vs. Buffered IO, as well as the most dominant workload in Android platform: SQLite insert/update and a write followed by fsync() call. Mobibench can also vary the number of concurrent threads to examine the storage and file system overhead to support concurrency, e.g., metadata updates, journal file creation/deletion. MOST capture the trace and extracts key file system access characteristics: access pattern with respect to file types, ratio between random vs. sequential access, ratio between buffered vs. synchronous IO, fraction of metadata accesses, etc. MOST implements reverse mapping feature (finding an inode for a given block) and retrospective reverse mapping (finding an inode for a deleted file). Research verified the performance result of Mobibench on eight smartphone models.

Developing complex applications for mobile platforms can become trivial for the Android platform. With all of the facility offered to the developer, it is necessary to evaluate the performance, the execution time and the power as the application is dissipating. Andrws Vieira et Al. [6] focused on the analysis of performance (execution time) and energy consumption of applications for Android in recursive and iterative algorithmic versions in the same computational complexity. The result of this analysis suggests the version of the algorithm that provides the best compromise between performance and power consumption for the platform.

Android applications normally exhibit poor responsiveness. The error "Application Not Responding" is common in these Apps whenever the user interface thread executes expensive operations. A systematic approach to address poor responsiveness in Android software is presented in the research [7]. The test cases of this research inserted long delays at typical suspected GUI operations and they observed increased response times for these GUI events. Expensive GUI operations are the reasons for poor responsiveness of Android apps as per their observations.

An approach for performance testing of mobile applications has also been proposed by researchers. Their aim is to assist mobile application developers to gauge performance requirements of the app from the end users' point of view [8]. In this research, for network emulation an in-house developed tool called WindTunnel has been used.

Android supports Dalvik Debug Monitoring System (DDMS) because of which profiling the performance becomes effective and easier. DDMS can be used to profile only a few performance parameters. It cannot be used o profile end-to-end performance of mobile applications. A cost-effective method is suggested in this paper for end-to-end performance analysis of mobile app. DDMS, New Relic performance profiler and Jmeter have been used for end – to-end performance analysis of android application.

3. METHODOLOGY

1. Create a online database in a registered server (Hostinger.in).
    - The database is created using MySQL in the server through "phpmyadmin" domain Store_data(data) & data_Store(Name, Url of data).
    - The data in the database usually contains the Url of the file to hosted and server contains the files.
2. Create an android apk (Scalibs) by using developer.android.tools.
    - Android transaction application like Scalibs uses client server communication to access and retrieve data using php files which are created for different client server related data operations.
3. Run the apk (scalibs) in android device.
    - The android application Scalibs is run on android device. On run the application queries the server through php file and intern the php file returns the data data_Store(Name, Url of data) queried by the server in an java script object notation(JSON object).
    - The data held by this object is decoded and displayed to client. In any successful transaction the client requests the server for service by registering to the server, in scalibs the client can download desired eBook by registering to the server through valid email address and other details.
    - When client completes the registration the application(Scalibs) posts the data using http client to the browser and calls the associated php file which copies the posted data and inserts that data entered into the database which is stored in the server Store_data(data).
4. Analyse the performance of Scalibs using DDMS and New Relics.
    - The performance of the application is monitored in real time.
5. Collect the results and optimize the application until the desired performance is achieved.

4. ANDROID PERFORMANCE ANALYSIS TOOLS

Profilers used for performance measurement are DDMS(Dalvik Debug Monitoring System) [9], New Relic Profiler [10] and JMeter [11].

4.1 DDMS

Android ships with a debugging tool called the Dalvik Debug Monitor Server (DDMS), which provides port-forwarding services, screen capture on the device, thread and heap information on the device, logcat, process, and radio state information, incoming call and SMS spoofing, location data spoofing, and more.

On Android, every application runs in its own process, each of which runs in its own virtual machine (VM). Each VM exposes a unique port to which a debugger can attach. When DDMS starts, it connects to Android Debug Bridge (ADB). When a device is connected, a VM monitoring service is created between ADB and DDMS, which notifies DDMS when a VM on the device is started or terminated. Once a VM is running, DDMS retrieves the VM's process ID (pid), via ADB, and opens a connection to the VM's debugger, through the ADB Daemon (ADBD) on the device. DDMS can now talk to the VM using a custom wire protocol.

In Android 4.0, the DDMS (Dalvik Debug Monitor Server) includes a Detailed Network Usage tab that makes it possible to track when our application is making network requests. Using this tool, we can monitor how and when our app transfers data and optimize the underlying code appropriately. We can also distinguish between different traffic types by applying a "tag" to network sockets before use.

Log Cat is integrated into DDMS When we have set up our logging; we can use the Log Cat feature of DDMS to filter certain messages with the following buttons:

- Verbose
- Debug
- Info
- Warn
- Error

We can also setup our own custom filter to specify more details such as filtering messages with the log tags or with the process id that generated the log message. The add filter, edit filter, and delete filter buttons let we manage our custom filters.

4.2 New Relic Profiler

New Relic Profiler is a software analytics Software by a company named New Relic based in San Francisco, California. New Relic's technology, delivered in software as a service (SaaS) model, monitors Web and mobile applications in real-time that run in cloud, on premise, or hybrid environments.

Its software analytics platform includes Android and iOS native mobile apps monitoring in real time.

4.3 Apache Jmeter

It is an open source performance evaluation tool for mobile and web services. It can be used to record transactions performed in a session using built in recorders. Each transaction can be recorded by creating a proxy on the device on which Jmeter is running. A record can be saved and can be simulated for many simultaneous users, hence the load on system can be evaluated for different number of users. The results are measured in terms of throughput and latency.

5. PERFORMANCE ANALYSIS OF SCALIBS

The application Scalibs is a kind of book store developed which is similar to Amazon bookstore, where user can preview the books listed on the server by providing their personal information such as name mobile number email id address and the quantity of books the customers wants to buy. This application gets connected to bookstore through wifi connectivity. Listing of books can be viewed either in grid view or in list view. There is also a search books option that can be used to search for a specific book.

Application is built user-friendly with special navigation drawer that slide in and slide out on swipe. Different operations like changing the views, filtering the data shown in the application as per user request through search interface and options which comprises of about and exit interfaces. Search View filters the data in database by search string entered by the user. Fig. 1 shows the page downloaded for the search item 'Jeffry Archer'.

Using Trace View of DDMS, the thread Concurrency for various activities such as uploading data to the server, receiving data from the server, network usage for loading different pages such as buy and information have been analyzed. The network-layer analysis summarizes the data communication of the app via WiFi or 3G. Fig. 2(a) shows Network Usage of Scalibs when first page is loaded. Different images are loaded from database to grid view. The speed of data transfer for different frames is plotted against time. It is seen that speed was almost 489KBps. Fig. 2(b) shows Network Usage of Scalibs when second page is loaded. Selected image is loaded from database to grid view. The speed of data transfer for different frames is plotted against time. It is seen that speed was almost 121KBps. Similarly, the network usage of other activities such as buy and information pages has been analyzed. It is found to have network usage rate is different for different activity.

Fig-1: Scalibs Search page

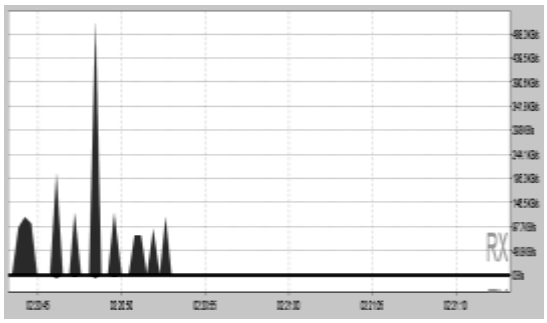


Fig-2(a) Network Usage for Loading First Page

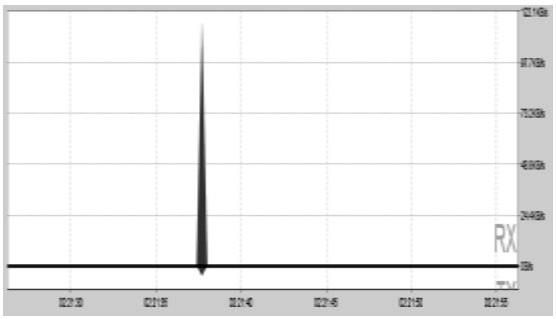


Fig-2(b) Network Usage for Loading Second Page

Fig. 3 shows Network Usage of Scalibs when information page is loaded. The speed of data transfer for different frames is plotted against time. It is seen that speed was almost 85KBps.

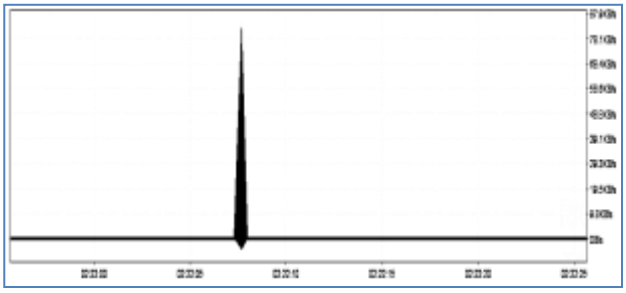


Fig-3: Network Usage for Information Page

Fig. 4 shows Network Usage of Scalibs when buy page is loaded. Data entered by the user is sent to the database. The speed of data transfer for different frames is plotted against time. It is seen that speed was almost 1.71KBps.

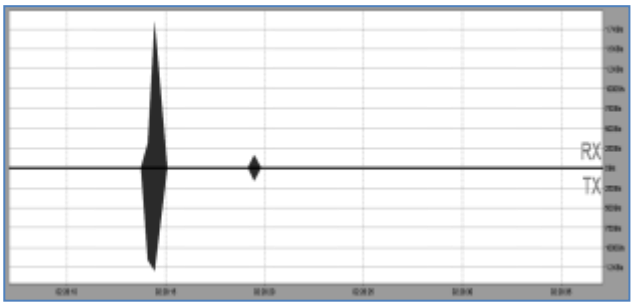


Fig-4: Network Usage for Buy Page

New Relic profiler is used to visualize the hotspots, execution time taken by each operation within the application such as image downloading, json script execution, the interaction between the activities of the application by average time, execution time, the type of devices executing the app and the underlying OS.

For example, fig. 5(a) and (b) shows the hotspots in the application. It has been observed that Main, Information and Buy activities of the application were executed for an average time of 2,820ms, 1,480ms and 101ms respectively in last 24 hrs, and graphically it is shown that Main and Information activities were executed thrice on April 2011 and Buy was executed only twice on the same day. Information page is identified as hotspot when compared to buy page as the application did not have the option for secure connection to payment gateway.

The experiments are conducted on two types of devices HTC phone and Asus Tablet and with different versions of OS. The average interaction time and Http response time of the application for different versions of the Android OS, and the usage of different carrier signals have been analyzed. The profiling results obtained are different for different devices.

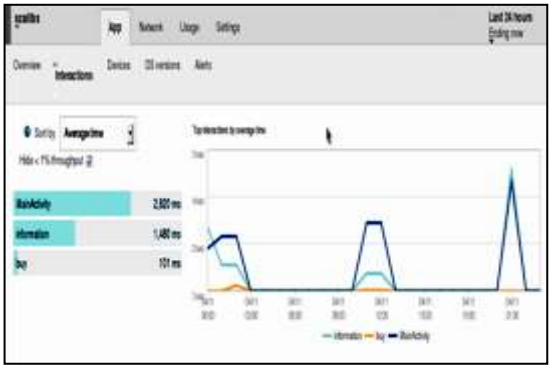


Fig-5(a): Hotspots Analysis

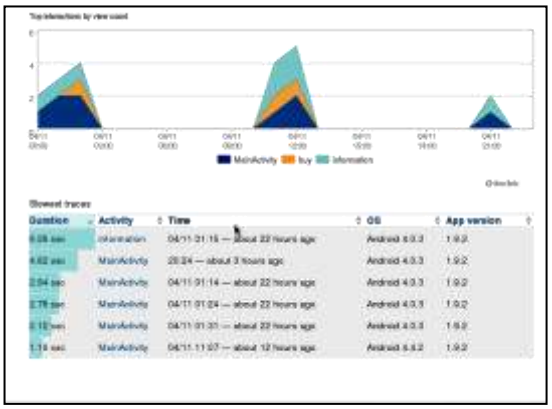


Fig-5(b): Activities Executed

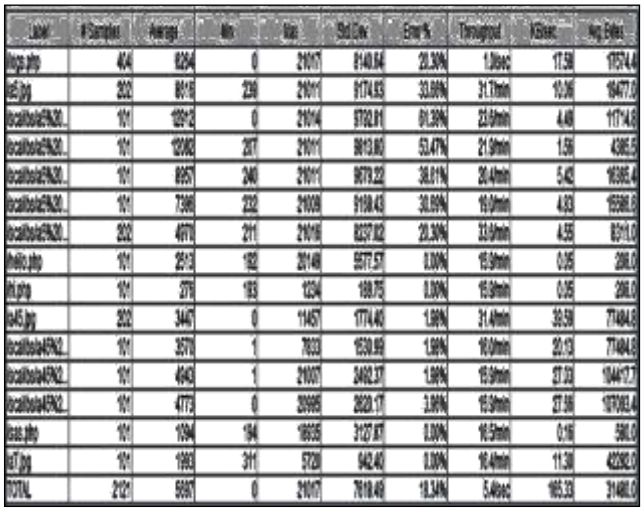
Apache JMeter, a cross platform application from the Apache Foundation has been used for load testing. JMeter is an open source desktop application. Load testing is conducted to check the volume of load the system can manage, and measure the response time. Multiple simultaneous users can be simulated for the application using JMeter. JMeter is used to analyze the following performance aspects of the application.

- Concurrency of the system; i.e., up to how many number of concurrent users the system is responsive.
- Throughput for given number of active users.
- Response time of components used in the application on load.

We made a set-up of our test plan to run for 100 users, with all users starting their test within 10 seconds, and have each user perform the recorded scenario 1 time. The resulting summary report is shown in table 1.

In Table 1, label is the name/URL for the specific HTTP(s) Request. Label can be prefixed by Thread Group. #Samples is the number of users per request. Average is the average time taken by all the samples to run a label. In our experiment, average time for first label is 6264 ms and total average time is 5697 ms. Min and Max are the minimum and maximum time taken by a sample for specific label, which is 0ms and 21017ms in our experiment. Std. Dev. is the set of cases deviating from the average value of sample response time. The lesser value of std. div. represents more consistent data. Error% is the percentage of failed requests/Label. Throughput is the count of requests processed/ seconds by the server. KB/Sec is the amount of data downloaded per second from the server during the performance test execution.

Table-1: Summary Report



The Analysis of profiling are as noted below

- Performance of GUI component is difficult to track as GUI design depends on functionalities provided by the Android framework, and the responsive design of the GUI makes some of the components shown or hidden depending upon the screen size.
- Response time depends on the inter-arrival delay of the events.
- Asynchronous events designed to handle external tasks may incur indefinite delays preventing execution of other events.
- Performance aspects are dependent on the type of device and the OS versions being used.

We made a set-up of our test plan to run for 100 users, with all users starting their test within 10 seconds, and have each user perform the recorded scenario 1 time. The resulting graph is as shown in Chart 1.

Chart 1 is self-explanatory, with lines representing the average, median, deviation, and throughput. Average, Median, and Deviation show average, median, and deviation of the number of samplers per minute respectively, while Throughput shows the average rate of network packets delivered over the network for our test run in bits per minute.

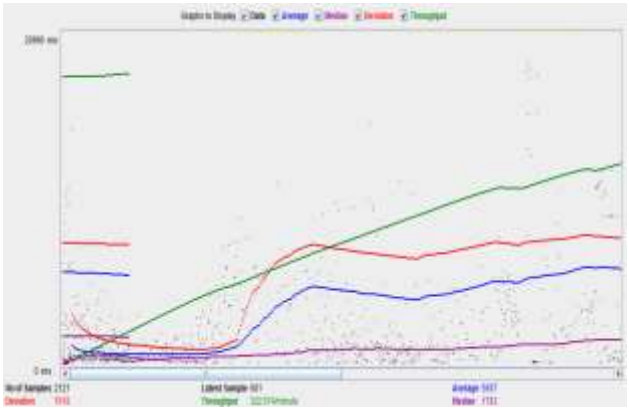


Chart-1: Samplers and Throughput for the Test scenario

6. CONCLUSION AND FUTURE WORK

All performance issues with the related data transactions between android application and online database are profiled and analysed. The profiling methodology is cost-effective as the profilers are open source profilers. Profiling results can be used to optimise the cloud based android application for better performance. Application Scalibs is optimized to achieve high performance using DDMS profiler and New Relic profiler. In future many more apps with varying design complexity and use cases will be profiled to analyse and optimise these applications for better performance. In future many more profiling tools will be studied and an attempt will be made to develop a profiling tool and analyse the performance of cloud based mobile applications using this tool.

REFERENCES

[1] Vijaya Shetty S, Dr. H. Sarojadevi, Navya K.M, Estimating Energy Usage of Transactions in Mobile Applications, SSRG- International Journal of Computer Science and Engineering, 2(7), (2015), 13-18.

[2] Vijaya Shetty S, Dr. H. Sarojadevi, e-Business Performance Issues, Quality Metrics and Development Frameworks, International Journal of Computer Applications, Foundations of Computer Applications, 55, (2012) , 42-47.

[3] Hyeon-Ju Yoon, "A Study on the Performance of Android Platform", ISSN: 0975-3397 Vol. 4 No. 04 April 2012, International Journal on Computer Science and Engineering (IJCSE)

[4] Xuetao Wei, Lorenzo Gomez, Iulian Neamtiu, Michalis Faloutsos "ProfileDroid: Multi-layer Profiling of Android Applications", 2012 ACM 978-1-4503-1159-5/12/08

[5] Sooman Jeong, Kisung Lee, Jungwoo Hwang, Seongjin Lee and Youjip Won, "AndroStep: Android Storage Performance Analysis Tool", Dept. of Electronics and Computer Engineering Hanyang University, Korea

[6] Andrws Vieira, Daniel Debastiani, Luciano Agostini, Felipe Marques, Júlio Mattos , "An analysis of power and performance of applications for mobile devices with Android OS", XXVII SIM - South Symposium on Microelectronics

[7] Shengqian Yang, Dacong Yan and Atanas Rountev , "Testing for Poor Responsiveness in Android Applications", Department of Computer Science and Engineering Ohio State University, 978-1-4673-6333-4/13/$31.00 c 2013 IEEE

[8] "Mobile Application Performance Testing: An End-2-End Approach", Infosys Research

[9] https://developer.android.com/develop/index.html

[10] "The New Relic", http://thenewrelic.com

[11] Bayo Erinle, "Performance Testing with JMeter 2.9", Open Source, Community Experience Destllied, PKCT Publishing.

BIOGRAPHIES

Dr. Vijaya Shetty S. is an Assoc. professor in the department of CSE at Nitte Meenakshi Institute of Technology (NMIT), Bengaluru, India. She is a PhD graduate from VTU. She has received her B.Tech. degree from Mangalore University and M.Tech. degree from VTU. Her research interests include Performance Engineering, Computer Architecture, Distributed Computing, Cloud Computing and IoT. She has published more than 20 papers in National and International conferences/journals and one patent. She has over 20 years of experience in academics and research.

Dr. Sarojadevi H. is a professor in the department of CSE at NMIT, Bengaluru, India. She is a PhD graduate from the Indian Institute of Science, Bangalore. She has received B.E. and M.E. qualifications from the University Visvesvaraya College of Engineering, Bangalore. She has published about 100 papers, 2 books and one patent. Her work experience of 23 years includes industry and academics. She has won several national and international recognitions, including Marquis Who's Who in the World 2016.